

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ХМЕЛІВСЬКИЙ ЮРІЙ СЕРГІЙОВИЧ

Допускається до захисту:

завідувач кафедри

інформаційних технологій,

канд. техн. наук, доцент

_____ Т. В. Нескородева

« _____ » _____ 2021р.

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ З ІГРИ

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Римар П. В., старший викладач

кафедри інформаційних технологій

Оцінка: _____/_____/_____

(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

(підпис)

Вінниця – 2021

АНОТАЦІЯ

Хмелівський Ю. С. Розробка мобільного додатку з іграми. Спеціальність 122 «Комп'ютерні науки», освітня програма «Сучасні інформаційні технології та програмування». Донецький національний університет імені Василя Стуса, Вінниця, 2021.

Кваліфікаційна (бакалаврська) робота присвячена розробці мобільного додатку-гри для платформи Android. У вступі наведено актуальність розробки, сформульована мета роботи, визначено об'єкт, предмет та задачі дослідження. Перший розділ присвячений постановці задачі та огляду існуючих аналогів. У другому розділі наведено стек технологій, які використовувалися для розв'язку задачі. Третій розділ присвячений опису розроблюваного додатку. У висновку підбитий підсумок виконаної роботи.

Ключові слова: Android, мобільний додаток, гра для вечірки.

Сторінок 42, рисунків 22, літературних джерел 21.

ABSTRACT

Khmelivsky Y. S. Development of a mobile application with games. Specialty 122 «Computer Science», educational program «Modern Information Technology and Programming». Vasyl' Stus Donetsk National University, Vinnytsia, 2021.

Qualifying (bachelor's) work is devoted to the development of a mobile game application for the Android platform. The introduction presents the relevance of the development, formulates the purpose of the work, defines the object, subject and objectives of the study. The first section is devoted to the problem statement and review of existing analogues. The second section describes the stack of technologies used to solve the problem. The third section is devoted to the description of the developed application. In conclusion, summed up the work done.

Keywords: Android, mobile application, party game.

Pages 42, figures 22, literature sources 21.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ІСНУЮЧИХ МОБІЛЬНИХ-ДОДАТКІВ ЗА ДАНОЮ ТЕМАТИКОЮ.....	6
1.1 Постановка задачі.....	6
1.2 Огляд існуючих додатків за даною тематикою	7
Висновок до розділу 1.....	13
РОЗДІЛ 2. ВИКОРИСТАНІ ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ.....	14
2.1 Платформа Android	14
2.2 Середовище для розробки Android Studio	16
2.3 Android SDK та Android API.....	17
2.4 Мова програмування Java.....	20
Висновок до розділу 2.....	27
РОЗДІЛ 3. РОЗРОБКА ДОДАТКУ «STUDENTS STUDIOS»	28
3.1 Структура проекту	28
3.2 Розробка додатку.....	29
3.2.1 Головне меню	29
3.2.2 Гра «Завдання»	30
3.2.3 Гра «Бомба»	33
3.2.4 Гра «Аліас»	35
3.3 Інсталяція та системні вимоги.....	37
Висновок до розділу 3.....	37
ВИСНОВКИ.....	39
СПИСОК ЛІТЕРАТУРИ.....	40

ВСТУП

Мобільні додатки увірвалися в життя людини всього десять років тому і майже відразу стали невід'ємною їх частиною. Без них неможливо уявити життя сучасної людини. Вони поділяються на декілька категорій за своїм призначенням, серед яких можна виділити ті, що будять людину вранці, допомагають підібрати одяг відповідно до погоди, показують час найближчого автобуса, повідомляють про нові листи на електронній пошті. За допомогою додатків можна шукати роботу, керувати своїми фінансами, слідкувати за подіями в світі та багато іншого.

Окрім додатків, які полегшують повсякденне життя людини, чинне місце на ринку мобільних додатків займають ігри[1]. Вони дозволяють цікаво провести вільний час і розважитись. Мобільний геймінг почав стрімко зростати у популярності і став в один ряд з найбільш розвиненими ігровими платформами, такі як ПК та консолі. Одним із жанрів мобільних ігор є настільні. До даної категорії відноситься гра, якій присвячена дана робота.

Мета бакалаврської роботи полягає у вивченні методики розробки мобільних додатків для платформи Andorid. Необхідно розібратися у можливостях, роботі та особливостях середовища розробки Android Studio, а також вивчити мову програмування Java[2].

Задачі дослідження полягають у:

- огляді існуючих додатків за даною тематикою;
- вивченні та вдосконаленні навичок використання спеціалізованого програмного забезпечення Android Studio для розробки додатку;
- розробці додатку з іграми.

Об'єктом для дослідження є ОС Android і середовище розробки Android Studio[3].

Предметом дослідження є технологія розробки додатку під операційну систему Android.

Робота складається зі вступу, трьох розділів, висновку та списку літературних джерел кількістю 21 найменувань та 22 рисунків. Загальний об'єм роботи 42 сторінки.



РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ІСНУЮЧИХ МОБІЛЬНИХ- ДОДАТКІВ ЗА ДАНОЮ ТЕМАТИКОЮ

Даний розділ присвячений постановці задачі, правилам гри, а також огляді існуючих додатків за даною тематикою.

1.1 Постановка задачі

Необхідно створити мобільний додаток для набору міні-ігор, використовуючи мову програмування Java[2]. Додаток створений для платформи Android версії 5.0 і вище[4]. Додаток має назву «Students studios».

Правила ігор:

1. «Завдання». В даній грі гравцям по колу буде випадково видане цікаве завдання, яке учасник гри має виконати. Якщо все ж таки він не бажає цього робити, то завдання можна пропустити і воно може випасти комусь пізніше. Також реалізована можливість повернення до попереднього завдання, якщо було випадково перейдено далі.
2. «Бомба». За певний час, який залежить від кількості гравців, кожен повинен якомога швидше назвати слово-відповідь для завдання на екрані. Після відповіді, хід переходить до наступного гравця за годинниковою стрілкою. Гра продовжується до тих пір, поки бомба не вибухне. Переможеним вважається гравець, під час якого вибухнула бомба. Він же починає наступний раунд.
3. «Аліас». Це словесна командна гра, в якій гравцю необхідно за деякий час пояснити, а решті команді – відгадати якомога більше слів. За кожне правильне слово команда отримує бал. Команда, яка набере найбільшу кількість балів – перемагає.

До додатку були висунуті вимоги до дизайн додатку, який повинен бути дружнім до користувача та додаток потрібно розробити адаптивним для пристрою з довільною діагоналлю екрана.

1.2 Огляд існуючих додатків за даною тематикою

За аналог було взято Android-додаток гри «Party Monster».



Рисунок 1.1 – Android-додаток «Party Monster»

Додаток має набір з кількох ігор, наприклад:

- «Круг смерті» – гравці по черзі виконують завдання.



Рисунок 1.2 – Гра «Круг смерті»

- «Дінь-дінь» - ведучий читає ситуацію, той, хто був в описаній ситуації, повинен загнути палець на руці, якщо загнуті 5 пальців гравцю дається каверзне питання і починає загинати пальці знову.

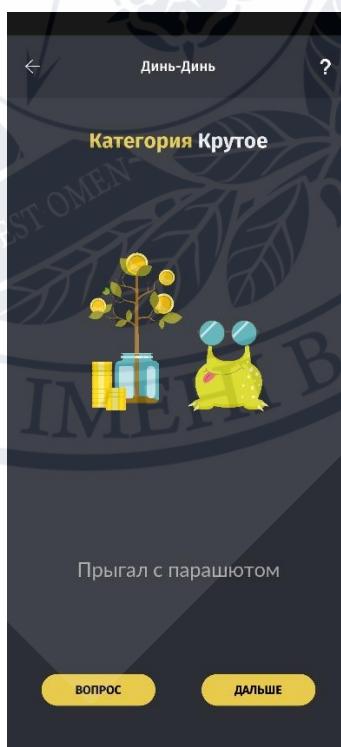


Рисунок 1.3 – Гра «Дінь-дінь»

- «Хто найбільш ймовірно» - після прочитання кожного твердження, кожен гравець називає когось з поміж гравців, хто на його думку більше відповідає твердженню. Людина, на яку вказала більшість має виконати завдання.



Рисунок 1.4 - Гра «Хто найбільш ймовірно»

- «Так чи ні» - ведучий зачитує загадку і дізнається відповідь, задача решти гравців її відгадати. Гравці можуть задавати питання, відповідь на які можуть бути «так» чи «ні».

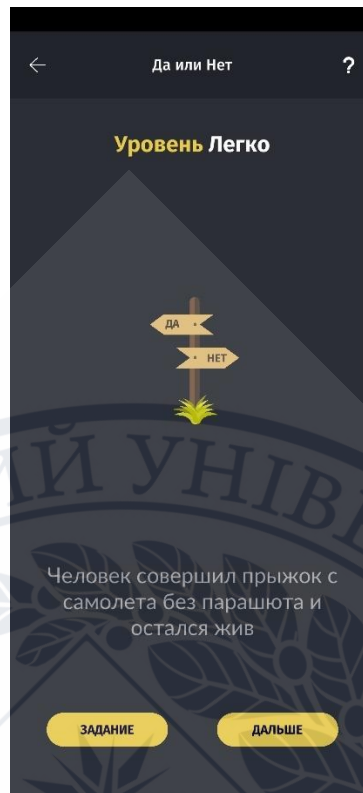


Рисунок 1.5 - Гра «Так чи ні»

- Імпровізація» - по черзі від двох команд відділяється один учасник кожен раунд. Решті команди задається певна ситуація, яку цей гравець має відгадати. Команда показує лише жестами, без слів.



Рисунок 1.6 – Гра «Імпровізація»

Також в додатку описаного в роботі, міститься гра «Бомба», аналогом якої є додаток «Bomb Party game». Ігри по функціоналу схожі, відрізняються лише дизайном.



Рисунок 1.7 – Android-додаток «Bomb Party game»



Рисунок 1.8 – Android-додаток «Bomb Party game»

Ще одна гра – «Alias», яка описана в роботі, реалізована в додатку, рисунок якого наведений нижче.

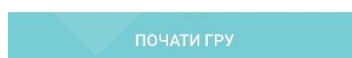
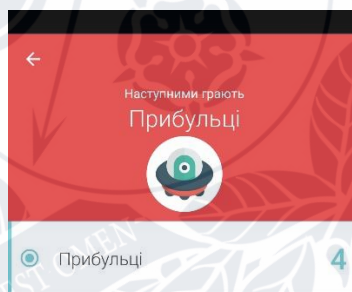


Рисунок 1.9 – Android-додаток «Alias 3»

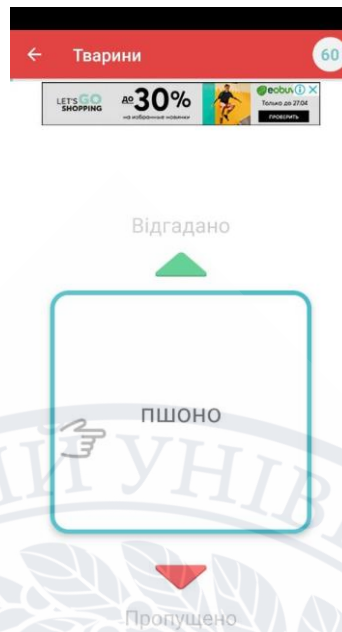


Рисунок 1.10 – Android-додаток «Alias 3»

Перевагою додатку «Students studios» є те, що в ньому зібрані всі переваги і особливості наведених аналогів, а також додані свої, такі як завдання для виконання, слова для відгадування та зрозуміліший і простіший у користуванні дизайн.

Висновок до розділу 1

В першому розділі описана постановка задачі, а також наведено аналогічні мобільні додатки, які було обрано за основу. Були окреслені їх плюси та мінуси. Наступний розділ буде присвячений вибору інструментів для розробки.

РОЗДІЛ 2

ВИКОРИСТАНІ ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ

В цьому розділі наведено стек технологій, які були використані для поставленої задачі з розробки мобільного додатку.

2.1 Платформа Android

Android – операційна система і платформа для мобільних телефонів, планшетів, електронних книг, навігаторів, смарт годинників, фітнес-браслетів, фотоапаратів, ігрових приставок, холодильників, нетбуків, окулярів Google Glass, телевізорів та інших пристроїв.

Платформа заснована на власній реалізації віртуальної машини Java, а також ядрі Linux. Операційною системою володіє Google, яка купила її в 2005 році в компанії Android.Inc., яку потім купила Google. Android відкриває можливість створенню Java-додатків, які в свою чергу через розроблені Google бібліотеки керують пристроєм.[5]

Чому саме Android платформа?

На даний час Android – найпопулярніша операційна система в світі. За офіційними даними агрегатора статистичних даних Statcounter, станом на кінець 2020 року, серед усіх пристроїв, як десктопних, так і мобільних, перше місце очікувано посідає операційна система Android – 39,77%, за нею ідуть Windows – 32,31%, IOS – 17,66% та OS X – 7,98%. Це не дивно, адже в світі людям простіше дозволити собі купити пристрій з ОС Android, ніж персональний комп'ютер або ноутбук, оскільки він в середньому дешевший, доступніший, зручніший у користуванні та й багатьом користувачам вистачає функціоналу виключно мобільного гаджета для повсякденних завдань. Серед мобільних телефонів Android охоплює значну частку в 70,43%, в той час як IOS – 29,06%. [6]

Також Android зручний і зі сторони розробника програмного забезпечення для мобільних пристроїв. Для програмування на Android є багато зручних середовищ розробки, такі як Android Studio, IntelliJ, NetBeans IDE, Visual Studio

та інші, які підтримують такі популярні мови програмування як Java, Kotlin, C# та інші. Кожен розробник може опублікувати свій додаток у Play Market та, через велику популярність платформи, мати досить велику кількість потенційних клієнтів, що принесе великий прибуток.

Отже, до числа переваг Android відносяться:

- Відкритий код – дана перевага дозволяє будь-якому кваліфікованому розробнику написати програму і поширити її в офіційний магазин ОС Android.
- Зручність передачі даних – при бажанні користувача можна завантажити файл будь-якого формату в пам'ять смартфона через USB-кабель.
- Ціна – можна знайти смартфон на Android, який буде по кишені кожному.
- Зручна синхронізація даних з електронними пристроями.
- Play Market – фірмовий магазин додатків, вагома частина яких є безкоштовною.[7]
- Можливість встановлення сторонніх програм, яких немає в офіційному магазині Play Market. В операційній системі Android є функція, яка дозволяє завантажувати додатки зі сторонніх джерел.
- Кастомізація. Адаптація і налаштування продукту під смаки індивідуального користувача дає можливість більш зручного користування пристроєм.

Поряд із численними перевагами, існують і певні недоліки, до числа яких можна віднести:

- Швидке витрачання заряду батареї.
- Якість заліза.
- Підтримка оновлень.
- Вразливість до атак хакерів.[8]

Отже, операційна система Android має як і переваги, так і недоліки, тому покупець смартфона може обирати пристрій з операційною системою, виходячи з особистих вимог та бюджету. Але як показує статистика, більшість надають перевагу саме ОС Android.

2.2 Середовище для розробки Android Studio

Інтегроване середовище розробки (IDE) під назвою Android Studio дозволяє працювати з платформою Android. IDE є офіційним засобом розробки Android додатків. Середовище розроблене на базі програмного забезпечення IntelliJ IDEA, що створене компанією JetBrains.[9]

Такі мови програмування як Java, Kotlin та C/C++ підтримуються для розробки мобільних додатків в Android Studio. Також програма містить велику бібліотеку шаблонів і компонентів та вбудований емулятор, що суттєво спрощує та прискорює розробку програмних продуктів.

В Android Studio можна створювати програми для останньої версії Android. Також функціонал середовища дає змогу одразу після створення додатку перевірити наявність помилок, різними інструментами протестувати додаток для виявлення багів та несправності в роботі. За допомогою вбудованого емулятора можна тестувати програму під пристрої на різних версіях системи та з різними параметрами та проводити оптимізацію для коректної роботи.

Розробники-початківці часто вибирають Android Studio, оскільки дане середовище більш зрозуміле і в ньому можна працювати навіть без особливого досвіду в програмуванні. Вагомою перевагою Android Studio є наявність функції SDK, яка значно спрощує локалізацію додатків.[10]

Недоліки

Android Studio є фактично ідеальною IDE для створення мобільних додатків під операційну систему Android, але для комфортної роботи в середовищі знадобиться потужний комп'ютер. Сучасні ПК часто повністю відповідають всім необхідним вимогам, а ось на старих машинах – можливі проблеми. Попри наявність вбудованого емулятора всередині середовища розробки, тестуванням програм можуть викликати труднощі. Оскільки для його запуску необхідні досить потужні можливості апаратної частини ПК, на якому планується тестування. Також недоліком є відсутність змоги написати на мові Java серверну частину для Android додатків. [11]

2.3 Android SDK та Android API

SDK (Software Development Kit) являє собою набір інструментів для створення програмного забезпечення під конкретну платформу. У загальному випадку в нього можуть входити компілятор, відладчик, емулятор, різні бібліотеки, інструменти, документи, приклади. Android SDK – це модулі коду на Java. Вони надають програмісту доступ до функцій мобільних пристроїв, стороннім сервісам і подібним речам.

Одним з ключових компонентів Android SDK є бібліотека Gradle. Для того, щоб інтегрувати в додаток платформу соціальних мереж (наприклад, Facebook), має бути завантажена бібліотека коду (або SDK) з Facebook, а потім повідомлено Gradle про це. Середній мобільний додаток для Android використовує близько 15 окремих SDK. Найпопулярнішими їх категоріями є аналітика і реклама.

Поліпшення Android SDK йдуть рука об руку з загальним розвитком платформи Android. SDK також підтримує старі версії Android, якщо розробники хочуть зосередити свої додатки на старих пристроях. Інструменти розробки – це завантажені компоненти, тому після завантаження останньої версії і платформи, старі платформи і інструменти можуть бути використані для тестування сумісності.[3]

Як правило, SDK містить API фреймворку, реалізацію API та різні системні образи для емулятора Android.

API (Application Programming Interface – програмний інтерфейс програми) – це набір чітко визначених методів, які забезпечують зв'язок між різними компонентами програмного забезпечення. API працює як договір між постачальником (наприклад, бібліотекою) та споживачем (наприклад, додатком).

API передбачає набір формальних визначень, серед яких класи, функції, методи, структури, константи які можуть бути використані іншими програмістами для написання власного коду. Проте API реалізується окремою програмною бібліотекою.

Платформа Android надає каркасний API, який програми можуть використовувати для взаємодії з базовою системою Android. Фреймворк API складається з:

- Основний набір пакетів та класів.
- Набір XML-елементів та атрибутів для оголошення файлу маніфесту.
- Набір XML-елементів та атрибутів для декларування та доступу до ресурсів.
- Набір інтентів.
- Набір дозволів, які можуть вимагати програми, а також засоби забезпечення дозволів, включені в систему.

Оновлення API фреймворку розроблено таким чином, щоб новий API залишався сумісним із попередніми версіями API. Тобто, більшість змін в API є адитивними та вводять або нові функції або ті, що їх замінюють. У міру оновлення частин API старі замінені частини застарівають, але не видаляються, так що існуючі програми все ще можуть ними користуватися. У нечисленній кількості випадків якісь частини API можуть бути або видалені, або замінені, хоча часто такі зміни необхідні тільки для забезпечення надійності API та безпечності додатків чи системи. Частини API, яких не торкнулося оновлення, переносяться без змін з попередніх версій.

Додатки можуть використовувати елемент маніфесту, наданий фреймворком API для рівнів, на яких вони можуть працювати, а також бажаний рівень API, який вони призначені для підтримки.

Як правило, проект під Android складається з коду, написаного розробниками з використанням Android API, а також деяких інших бібліотек, залежностей і ресурсів.[13]

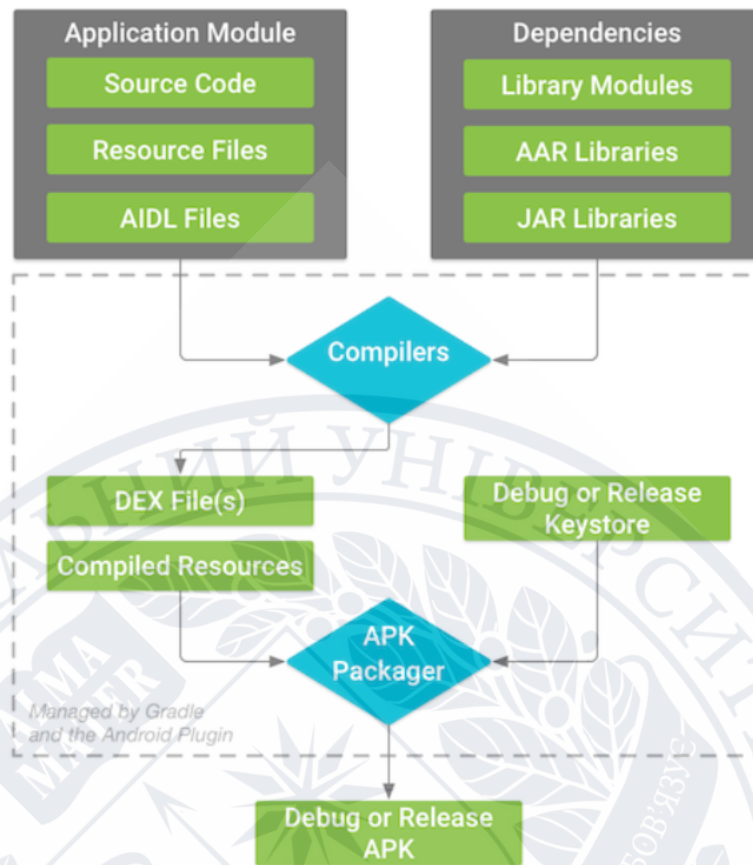


Рисунок 2.1 – Процес збірки файлу APK

Процес побудови типового модуля програми для Android, як показано на Рисунок 2.1, виконує такі загальні кроки:

1. Компілятори перетворюють початковий код, написаний на Java або Kotlin, включаючи залежності, у скомпільовані ресурси та файли DEX, які містять байт-код, що працює на пристроях під Android.
2. Пакувальник APK з'єднує файли DEX та скомпільовані ресурси в єдиний файл APK.
3. Пакувальник APK підписує ваш APK, використовуючи налагодження (debug) або зберігання (release) ключів:
 - Якщо створюється налагоджувальна версія додатка, тобто додаток, який призначений тільки для тестування, пакувальник підписує додаток магазином ключів налагодження.
 - Якщо створюється випускна версія додатка, то пакувальник підписує програму сховищем ключів випуску.

4. Останнім етапом створення файлу APK є використання інструменту `zipalign`, який оптимізує додаток, щоб під час роботи додатка на пристрої використовувалося менше ресурсів пам'яті.

Кінцевим результатом процесу збірки є файл формату `.apk` для налагодження або для випуску, який можна використовувати для розгортання, тестування або випуску для зовнішніх користувачів.[12]

Android Runtime

Android Runtime (ART) – це керований час виконання, який використовується програмами та певними службами ОС Android. ART під час виконання виконує формат Dalvik Executable та специфікацію байт-коду Dex.

ART та Dalvik – сумісні середовища виконання під керуванням байт-коду Dex, тому програми, розроблені для Dalvik, повинні працювати під час роботи з ART. Однак деякі методики, які працюють над Dalvik, не працюють над ART.[12]

Gradle

Gradle – це система автоматичного складання. В Android Studio Gradle супроводжує розробника під час роботи постійно. Система Gradle використовується не лише в мобільній розробці, а й в додатках написаних на інших мовах та під інші платформи.[14]

2.4 Мова програмування Java

Java – об'єктно-орієнтована мова програмування. З 2009 року мова відійшла під опікування компанії «Oracle» [15]. В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні за допомогою віртуальної машини інтерпретується під конкретну платформу.

Java використовується дуже часто, вона є однією з мов програмування, на якій пишуть програми для ОС Android. Вона використовується в веб-додатках, веб-сайтах, десктопних додатків і технологіях обробки великих даних. Java підходить і для наукових проєктів. Ще за довго до появи смартфонів мова Java займала першість в програмуванні для мобільних пристроїв. Java

використовувалася для написання перших ігор для мобільних пристроїв. Завдяки своїй довгій історії, вона заробила своє місце в Залі слави програмування.

Плюси програмування на Java

Java - вже не єдина офіційно підтримувана мова для розробки на Android. Java далеко не єдиний вибір в веб-програмуванні. Проте, Java йде в ногу з часом. Давайте розглянемо, які переваги пропонує Java.

- Об'єктно-орієнтоване програмування.
- Java – мова високого рівня.
- Безпека.
- Крос-платформність.
- Автоматичне управління пам'яттю
- Багатопоточність. Між потоками можна швидко перемикатися, адже вони використовують одну і ту ж область пам'яті. Потоки незалежні один від одного: один потік не впливає на роботу інших потоків.

Мінуси програмування на Java

- Продуктивність дуже низька.
- Немає нативного дизайну.
- Для комерційного використання необхідно офіційно купувати ліцензію.

Java, як і раніше, залишається фактично основною мовою Android-додатків, хоча зараз активно зростає популярність мови Kotlin. Таким чином, всі програмісти, які знають Java дуже легко можуть стати Android-розробниками. Хоча Android використовує Android SDK замість JDK, проте, код написаний на Java [16].

Java чи Kotlin

Питання полягає в тому, чи варто розробнику переходити на Kotlin [17] з Java чи ні? Звичайно це залежить від уподобань розробника. Однак, перш ніж перемикатися, важливо зрозуміти різницю між двома мовами програмування.

- Документація

Хоча обидві мови мають достатньо документації для вивчення, простіше знайти підручник або книгу для Java, оскільки вона використовується вже довгий час.

- Виключення.

Відсутність умов для перевірки винятків у Kotlin є однією з основних відмінностей її від Java. Якщо програміст, що працює на Java, вважає неприємним використання блоків `try / catch` у коді, то упушення, яке зроблене в Kotlin, можна вважати перевагою. Це також можна трактувати і у зворотному напрямку.

- Стилість коду.

Kotlin є суттєво лаконічним, це можна побачити, якщо порівняти певну частину програмного коду написаного на Java з еквівалентним кодом написаного на Kotlin.

Це явно можна віднести до плюсів Котліна.

- Співпрограми або корутіни.

Процеси інтенсивно завантажують процесор і мережевий вхід-вихід, тому що це зазвичай тривалі операції. Оскільки Android за замовчуванням є однопоточним, призначений для користувача інтерфейс програми повністю зависне і не буде відповідати доти, поки операція не завершиться.

В Java рішення традиційно полягала в створенні фонових потоків, в якому можна було б виконувати цю інтенсивну або тривалу роботу, але управління декількома потоками може привести до складного, схильного до помилок коду, а створення нового потоку – дорога операція.

Хоча і можна створювати додаткові потоки в Kotlin, але доцільніше використовувати співпрограми або корутіни.

Співпрограми виконують тривалі й інтенсивні завдання, припиняючи виконання в певний момент, не блокуючи потік, а потім відновлюючи цю функцію пізніше, можливо, в іншому потоці. Це дозволяє створювати неблокуючий асинхронний код, який виглядає синхронним і, отже, більш чітким, коротким і легким для читання.

Більш того, співпрограми також не мають стеків, тому вони споживають менше пам'яті порівняно з потоками і відкривають двері для додаткових стилів асинхронного неблокуючим програмування, таких як `async/await`.

- Класи даних.

У проекті немає нічого незвичайного в наявності декількох класів, які нічого не роблять, крім зберігання даних. В Java спостерігається написання значної кількості шаблонного коду для цих класів, хоча самі класи мають дуже невелику функціональність.

Як правило, розробник повинен визначити конструктор і кілька полів для зберігання даних, функції геттери і сеттери для кожного з полів, а також функції `equals()`, `hashCode()` і `toString()`.

У Kotlin, якщо розробник допише тільки ключове слово `data` в визначення класу, компілятор виконає всю цю роботу сам, включаючи створення всіх необхідних геттерів і сеттерів. Зрозуміло, що це є очевидним плюсом для Котліна.

- Функції вищого порядку і лямбда.

Функція вищого порядку – це функція, яка або повертає функцію або приймає інші функції в якості параметрів. Також функції Kotlin є функціями першого класу. Це означає, що вони можуть зберігатися в змінних і структурах даних, що передаються в якості аргументів і повертають з інших функцій більш високого порядку.

З цього слідує, що у взаємодії з іншими нефункціональним значенням функції можуть працювати усіма можливими способами.

Kotlin використовує ряд функціональних типів для подання функцій, оскільки вона є статично типізованою мовою програмування. Більш того, цей ряд поставляється з набором лямбда-виразів – спеціалізованих мовних конструкцій.

Анонімні функції і лямбда-вирази також відомі як функціональні літерали. Це функції, які не оголошені, але передаються як вираження.

- Неявні розширюючі перетворення.

Kotlin не підтримує неявні розширюють перетворення для даних, тому менші типи не можуть бути перетворені в більші типи. У Kotlin, якщо ви хочете привласнити значення типу Byte змінної Int, вам потрібно буде виконати явне перетворення, тоді як Java підтримує неявні перетворення. Численна доля розробників сприймає це як мінус Котліна.

- Вбудована підтримка делегування.

У термінології програмування, делегування являє собою процес, в якому один об'єкт делегує свої операції іншому об'єкту-делегату. Kotlin підтримує шаблон проектування композиції поверх спадкування за допомогою делегування першого класу, також відомий як неявне делегування.

- Неприватні поля.

Задля досягнення певного рівня керованості в програмі необхідна інкапсуляція. За її допомогою, виходячи з того як блоки взаємодіють з об'єктом, можна встановити його уявлення. До того ж, якщо публічний API залишається незмінним, відкривається можливість змінити уявлення без необхідності зміни абонентів.

Інкапсуляція необхідна в будь-якій програмі для досягнення бажаного рівня керованості. За допомогою інкапсуляції, уявлення об'єкта може бути встановлено виходячи з того, як блоки взаємодіють з ним. Крім того, можна змінити уявлення без необхідності зміни абонентів, якщо публічний API залишається незмінним.

В Java Public поля корисні в сценаріях, де об'єкти, що викликають, повинні змінюватися відповідно до їх поданням. У Kotlin немає неприватних полів.

- Kotlin за замовчуванням є нульовим.

NullPointerExceptions - величезне джерело розчарування для розробників Java. Java дозволяє привласнити значення null будь-якої змінної, але якщо ви спробуєте використовувати посилання на об'єкт, що має нульове значення, приготуйтеся до виникнення виключення NullPointerException!

У Kotlin всі типи за замовчуванням не допускають значення NULL (не можуть містити значення NULL). Якщо відбудеться спроба присвоїти або

повернути null в своєму коді Kotlin, він завершиться помилкою під час компіляції.

Таким чином, в Kotlin немає винятків NullPointerException. Якщо зустрічається такий виняток в Kotlin, то, швидше за все, було або явно привласнено значення null, або це пов'язано з якимось зовнішнім Java-кодом.

- Примітивні типи.

Існує 8 примітивних типів даних, а саме char, double, float, int, short, long, byte та boolean. На відміну від Kotlin, в Java змінні примітивного типу не є об'єктами. Отже, що вони не є об'єктами, створеними з класу або структури.

- Розумні приведення.

Перш ніж об'єкт може бути приведений в Java, обов'язково потрібно перевірити тип. Це також вірно в сценаріях, де очевидно потрібно приводити об'єкт. На відміну від Java, Kotlin має функцію розумного приведення, яка автоматично обробляє такі надлишкові приведення. Вам не потрібно виконувати приведення всередині оператора, якщо він вже перевірений оператором is в Kotlin.

В Java часто доводиться перевіряти тип і потім приводити об'єкт в ситуаціях, коли вже ясно, що об'єкт може бути приведений.

Інтелектуальні перетворення Kotlin обробляє кожна надлишкові перетворення для вас, тому вам не потрібно виконувати приведення всередині оператора, якщо ви вже перевірили його за допомогою оператора Kotlin 'is'.

- Статичні члени.

В мові програмування Java за допомогою ключового слова static можна оголосити те, що конкретний член з модифікатором static, належить саме типу, а не екземпляру цього типу. З цього слідує, що створюється лише один екземпляр цього статичного члена, а також він дозволяє використовувати себе всім екземплярам класу.

У Kotlin відсутні статичні елементи.

- Підтримка конструкторів.

На відміну від Java, в Kotlin клас може мати первинний конструктор і один або кілька вторинних конструкторів, які створюються шляхом включення їх в оголошення класу.

- Трійчастий оператор.

В Kotlin відсутній тернарний оператор, який присутній в Java. Він функціонує подібно до базового оператора if. Його структура складається з умови, яка може бути істинною або помилковою.

- Бібліотеки обробки анотацій з Kotlin.

Kotlin підтримує всі існуючі фреймворки і бібліотеки Java, включаючи розширені фреймворки, які покладаються на обробку анотацій, хоча деякі бібліотеки Java вже надають розширення Kotlin, такі як RxKotlin.

Якщо виникає потреба використовувати бібліотеку Java, яка покладається на обробку анотацій, то її додавання в проект Kotlin трохи відрізняється, оскільки потрібно вказати залежність за допомогою плагіна kotlin-kapt, а потім використовувати інструмент обробки анотацій Kotlin (kapt) замість annotationProcessor.

- Взаємозамінність з Java.

Очевидно, що деякі моменти краще реалізовані в Kotlin, в той час як для інших – вигідно використовувати Java.

Незалежно від всіх відмінностей між двома мовами програмування, вони повністю сумісні. І Java, і Kotlin компілюються в байт-код. Це означає, що в питаннях «Kotlin vs Java» не буде однозначної відповіді, адже можна викликати код Java з Kotlin і навпаки.

Ця гнучкість має дві переваги. По-перше, це полегшує початок роботи з Kotlin, поступово додаючи код Kotlin в проект Java. По-друге, обидві мови можуть використовуватися одночасно в будь-якому проекті розробки додатків для Android.[18]

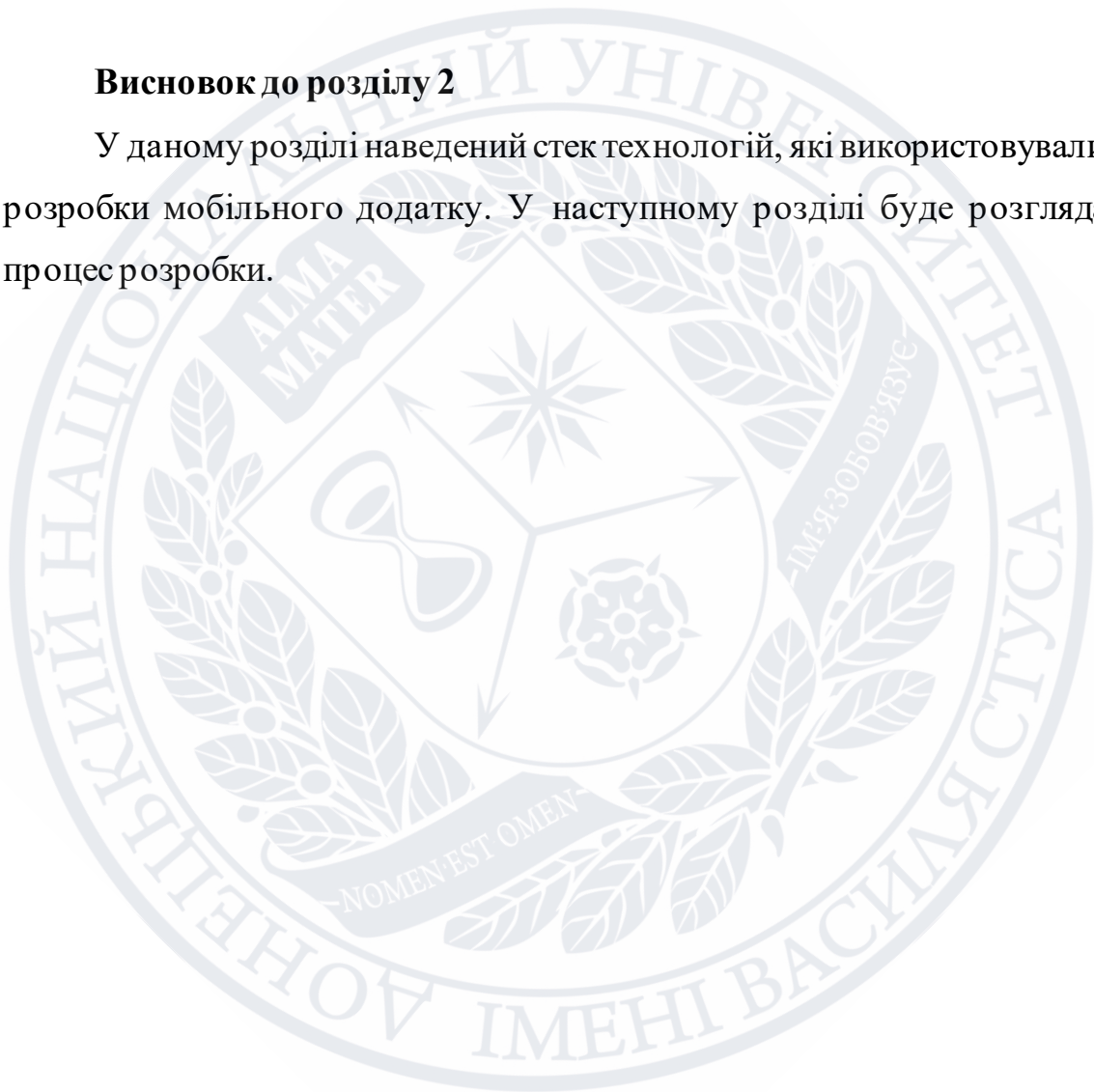
Отже, не звертаючи уваги на значні переваги Kotlin, в питаннях розробки загального призначення Java переважає конкурента. З іншого боку, все більше

розробників і організацій впроваджують Kotlin для швидкої розробки Android додатків.

Швидше за все, дебати про Kotlin і Java не врегульовуються найближчим часом, оскільки обидва мають свої переваги. Питання вибору: «чому Java?», «Чому Kotlin?» все ще залишаються. У будь-якому випадку перехід з Java на Kotlin НЕ буде болючим.

Висновок до розділу 2

У даному розділі наведений стек технологій, які використовувалися під час розробки мобільного додатку. У наступному розділі буде розглядатися сам процес розробки.



РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ «STUDENTS STUDIOS»

Даний розділ присвячений розробці мобільного додатку з іграми.

3.1 Структура проекту

Додаток гри складається з декількох вікон. Головним є меню з вибором конкретної гри. Також кожна гра має декілька вікон. Ігри «Бомба» та «Аліас» мають по одному додатковому модулю представлений типом Enum, вони містять список іменованих констант. Вибрана структура проекту є простою та інтуїтивно зрозумілою для користувача завдяки чіткій та зрозумілій організації модулів.

Рисунок 3.1 містить схему з програмним модулями. [19]

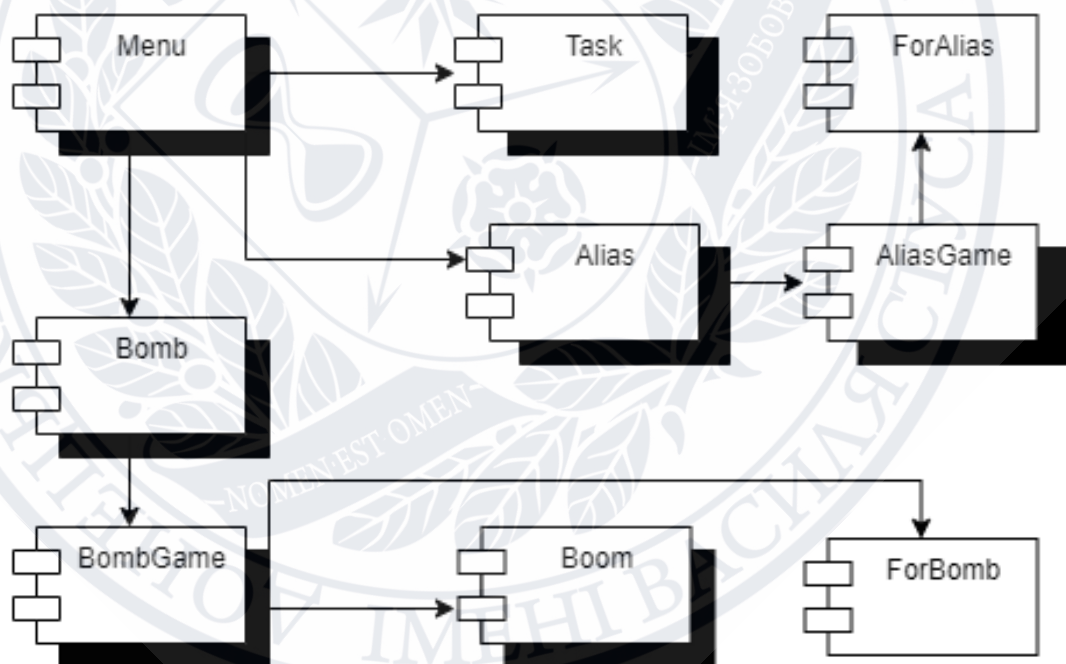


Рисунок 3.1 – Діаграма модулів

На рисунку 3.2 представлена діаграми роботи додатку та як вікна взаємодіють між собою[20].

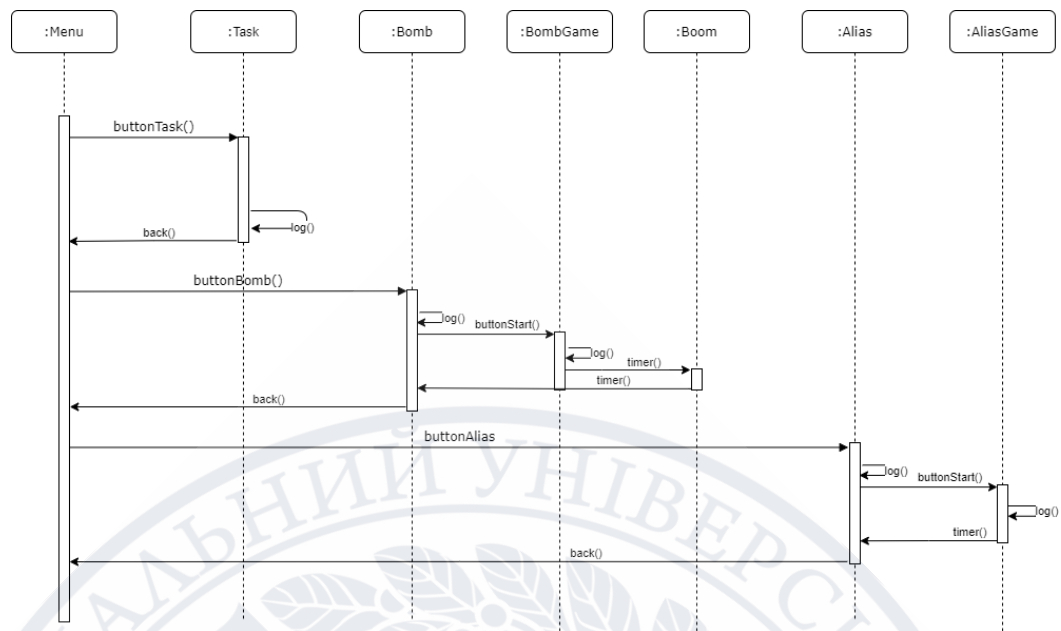


Рисунок 3.2 – Схема роботи додатку

3.2 Розробка додатку

Додаток створений за допомогою бібліотеки AppCompatActivity[21]. Вона забезпечує підтримку панелі додатків та інших можливостей на пристроях Android 2.1 і вище.

3.2.1 Головне меню

Головне меню має чотири елемента:

- Кнопка для входу в гру «Завдання»;
- Кнопка для входу в гру «Бомба»;
- Кнопка для входу в гру «Аліас»;
- Кнопка для виходу з додатку.

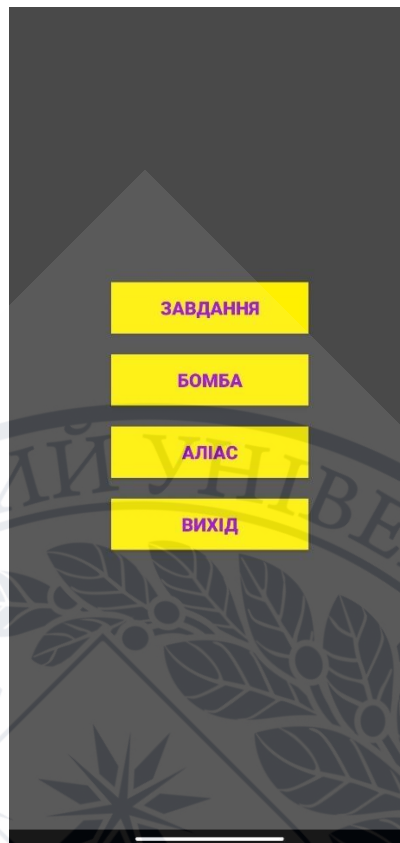


Рисунок 3.3 – Головне меню

3.2.2 Гра «Завдання»

Перша гра «Завдання». Вікно даної гри складається з декількох елементів:

- Область відображення картинок.
- Кнопка «Нове завдання».
- Кнопка «Пропустити».
- Стрілка повернення в меню.
- Лічильник зображень.
- Кнопка «-».
- Кнопка «+».

Випадковим чином з набору зображень на екран виводиться картинка з текстом завдання. При натисканні кнопки «Нове завдання» дане завдання вважається виконаним і воно видаляється з початкового набору зображень і більше не з'явиться, поки гра не буде заново запущена, а натомість відображається нове. Кнопка «Пропустити» переходить до наступного випадкового зображення, але не видаляє те, що було, тому воно з'явиться пізніше

в ході гри. Кнопка «-» дозволяє перейти до попереднього завдання, якщо воно було випадково пропущено. А кнопка «+» - переходить з попереднього до поточного. Стрілка у верхньому лівому кутку повертає користувача в головне меню. При переборі всіх зображень гра виходить в головне меню та її можна почати спочатку.

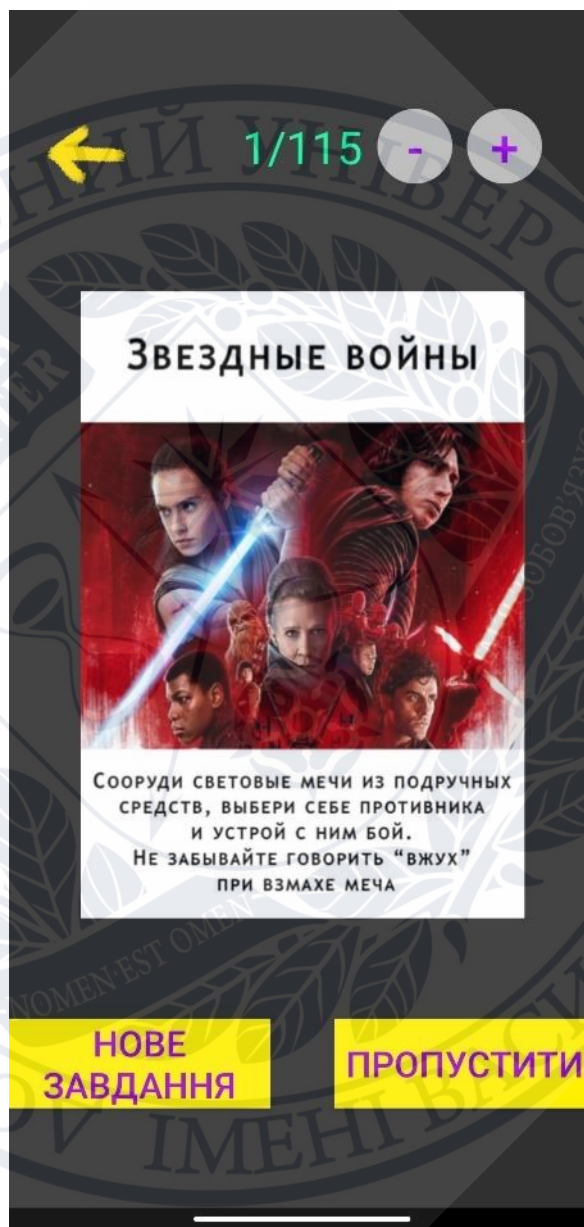


Рисунок 3.4 – Гра «Завдання»

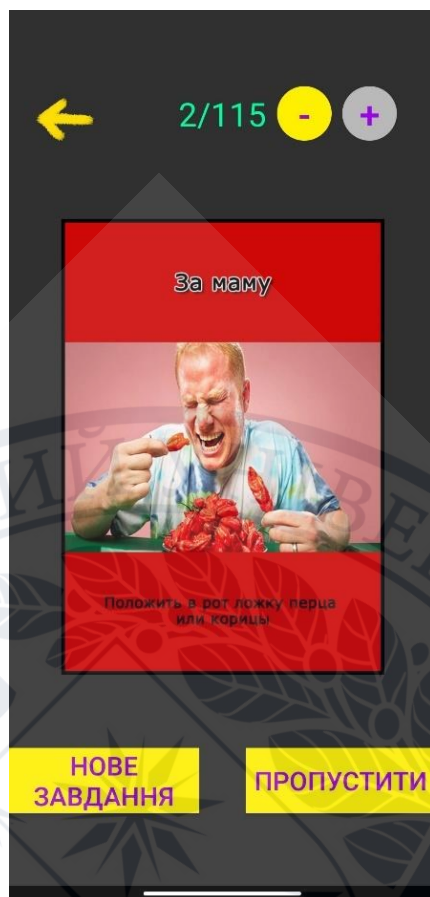


Рисунок 3.5 – Гра «Завдання»

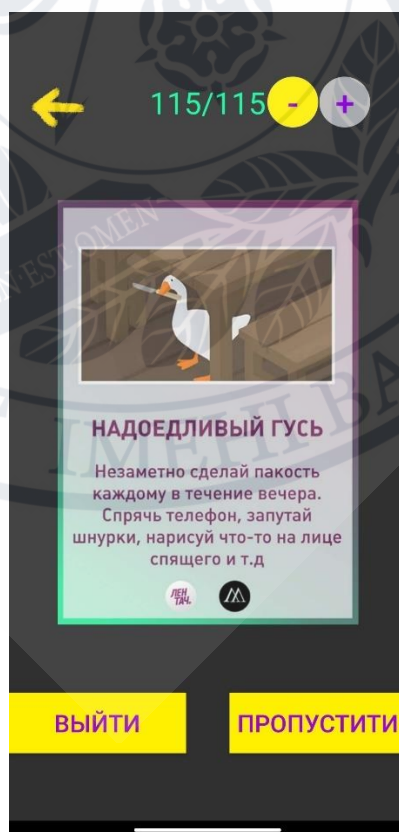


Рисунок 3.6 – Гра «Завдання»

3.2.3 Гра «Бомба»

Друга гра «Бомба». Вона містить кілька вікон. Перше вікно налаштування, воно складається з таких елементів:

- Кнопка «Старт».
- Лічильник кількості гравців.
- Кнопка повернення в меню.
- Зображення «бомби».

Друге вікно – процес самої гри. Воно містить наступні елементи:

- Область для виводу завдання.
- Кнопка повернення до попереднього вікна (у разі екстреного виходу до закінчення таймера).
- Зображення «бомби».

Третє вікно з'являється на кілька секунд для відображення «вибуху».

Перед початком запуску «бомби» визначається кількість гравців, які будуть грати в гру. Кількість гравців впливає на час, через який «бомба вибухне», тобто вичерпається час на раунд. Час розраховується в секундах за формулою: $5 + (5 * \text{«кількість гравців»})$. Натискаючи кнопку «Старт», запускається таймер і на екрані з'являється завдання. Коли час вичерпався, з'являється «вибух» і гра автоматично переходить на початковий екран гри.



Рисунок 3.7 – Гра «Бомба»



Рисунок 3.8 – Гра «Бомба»



Рисунок 3.9 – Гра «Бомба»

3.2.4 Гра «Аліас»

Третя гра «Аліас». Перше вікно призначене для відображення результату гри. Воно складається з таких елементів:

- Область відображення поточного рахунку гри;
- Кнопка «Старт»;
- Кнопка повернення в меню.

Друге вікно – процес гри, яке містить наступні елементи:

- Поле для відображення слів.
- Кнопка «Відгадано».
- Кнопка «Пропущено».
- Кнопка повернення до попереднього вікна (при цьому прогрес раунду для команди втрачається).
- Лічильник балів за даний раунд та назва команди.

Під час гри, якщо команда відгадала слово, натискається кнопка «Відгадано» і команді нараховується бал та видається наступне слово. Якщо ж команда не в змозі відгадати слово – кнопка «Пропущено» видає інше слово і за це знімає бал з команди. Після вичерпання часу, з'являється поточний результат гри для двох команд. І кнопкою старт знову запускається таймер, тільки вже бали нараховуються для іншої команди. І так по черзі між двома командами.



Рисунок 3.10 – Гра "Аліас"

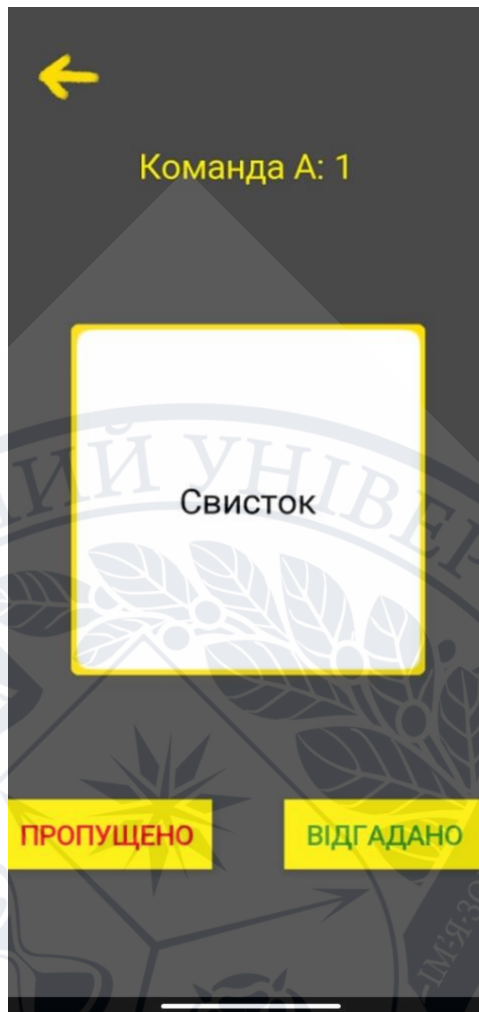


Рисунок 3.11 – Гра «Аліас»

3.3 Інсталяція та системні вимоги

Додаток інсталюється на мобільний пристрій за допомогою файлу формату .apk, після цього відбувається його автоматичне налаштування. Користувач повинен мати операційну систему Android на своєму пристрої, адже додаток розроблений під цю платформу. Для коректної роботи додатку необхідно мати версію операційної системи не нижче 5.0.0., проте найкомфортнішою для додатку буде десята версія операційної системи Android. Додаток не потребує підключення до мережі Інтернет чи будь-яких інших дозволів від користувача пристрою.

Висновок до розділу 3

Даний розділ містить правила кожної гри, яка міститься в додатку. Наведені екрани кожної з цих ігор.

Також приведена діаграма модулів даного додатка та схема його роботи. Описані системні вимоги до пристрою, на який буде встановлюватися додаток.



ВИСНОВКИ

В ході виконання бакалаврської роботи була докладно вивчена технологія створення мобільного додатку, а саме методи та засоби програмної реалізації продукту. Як наслідок, було обрано створення додатку під платформу Android. Для цього вивчено роботу інтегрованого середовища розробки Android Studio. А також були вдосконалені вже набуті навички та отримані нові знання при використанні мови програмування Java. Також було розглянуто схожі Android-додатки і перераховано їх плюси та мінуси.

Основним етапом у вивченні технології створення Android додатку стало написання повноцінного Android додатку з іграми «Students studios». Створений додаток відповідає всім сучасним нормам та має дружній дизайн. Він поєднує в собі зручність, функціональність і відповідну тематику. Додаток поєднує в собі набір ігор для вечірок в компанії друзів. Кожна міні-гра реалізована згідно загальних правил, які враховані при створенні додатку. У додатку зібрані всі переваги і особливості аналогів, а також додані свої, такі як завдання для виконання, слова для відгадування та зрозуміліший і простіший у користуванні дизайн. Гра є адаптивною для всіх пристроїв.

Android Studio є одним з основним середовищем розробки, і це не дивно, адже воно містить усі необхідні функції та можливості для зручного програмування додатків під Android, а також відповідає всіх сучасним вимогам та тенденціям розробки мобільних додатків. На даний момент Java залишається основною мовою програмування для платформи Android, хоча і стрімко розвиваються альтернативи такі як Kotlin.

Тестування даного програмного продукту показало, що додаток виконує покладені на нього функції, стабільно працює та загалом придатний для використання.

Отже, всі поставлені задачі було виконано в повному об'ємі та програма функціонує відповідно до правил гри. Також було отримано та засвоєно нові знання з розробки додатків під мобільну платформу Android.

СПИСОК ЛІТЕРАТУРИ

1. История и эволюция мобильных приложений [Электронный ресурс] – Режим доступа до ресурсу: <https://sdvv.ru/articles/testovyy-razdel/istoriya-i-evolyutsiya-mobilnykh-prilozheniy/>
2. Шесть основных языков программирования для мобильного разработчика [Электронный ресурс] – Режим доступа до ресурсу: <https://www.itweek.ru/mobile/article/detail.php?ID=175207>
3. Разработчик мобильных приложений [Электронный ресурс] – Режим доступа до ресурсу: <http://buduguru.org/profession/35>
4. Рейтинг лучших операционных систем для мобильных устройств [Электронный ресурс] – Режим доступа до ресурсу: <https://xiaomi-fun.ru/smartfony/mobilnye-operacionnye-sistemy.html>
5. Топ-10 Операционные системы мобильных телефонов [Электронный ресурс] – Режим доступа до ресурсу: <http://azericms.com/mobilnyie-prilozheniya/top-10-mobilnyih-telefonov-operatsionnyie-sistemyi.html>
6. Самые популярные операционные системы мира 2020 [Электронный ресурс] – Режим доступа до ресурсу: <https://marketer.ua/stats-operating-system-2020/>
7. Все приложения в Google Play безопасны: правда или миф? [Электронный ресурс] – Режим доступа до ресурсу: <https://www.kaspersky.ru/blog/google-play-malware/23629/>
8. Android vs iOS — что лучше, или почему я перешел с Galaxy на iPhone [Электронный ресурс] – Режим доступа до ресурсу: <https://icookie.ru/android-vs-ios/>
9. Початок роботи з Android Studio [Электронный ресурс] – Режим доступа до ресурсу: <https://code.tutsplus.com/uk/tutorials/getting-started-with-android-studio--mobile-22958>
10. Топ-3 среды разработки для Android [Электронный ресурс] – Режим доступа до ресурсу: <https://app-android.ru/blog/environment-develop-android>

11. Android studio документация на русском [Электронный ресурс] – Режим доступа до ресурсу: <https://dudom.ru/kompjutery/android-studio-dokumentacija-na-russkom/>
12. Android for Developers [Электронный ресурс] – Режим доступа до ресурсу: <https://developer.android.com/guide/>
13. API [Электронный ресурс] – Режим доступа до ресурсу: <https://blog.ingate.ru/seo-wikipedia/api/>
14. Building a web application with Gradle [Электронный ресурс] – Режим доступа до ресурсу: <https://openliberty.io/guides/gradle-intro.html>
15. Джава для початківців [Электронный ресурс] – Режим доступа до ресурсу: <https://lviv.logos-academy.com/chy-var-to-pochynaty-navchannya-na-programista-z-movy-java>
16. Почему Java? [Электронный ресурс] – Режим доступа до ресурсу: <https://art-code.com.ua/pochemu-java.html>
17. Kotlin Android Tutorial [Электронный ресурс] – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/kotlin-android-tutorial/>
18. Kotlin против Java – Сравните различия между Java и Kotlin [Электронный ресурс] – Режим доступа до ресурсу: <https://cutt.ly/HnmtMHH>
19. Диаграмма компонентов UML [Электронный ресурс] – Режим доступа до ресурсу: <https://planerka.info/item/diagrammy-komponentov-uml/>
20. Учебное пособие по диаграммам последовательностей [Электронный ресурс] – Режим доступа до ресурсу: <https://cutt.ly/2nmiBkR>
21. AppCompat [Электронный ресурс] – Режим доступа до ресурсу: <https://developer.android.com/jetpack/androidx/releases/appcompat>

Декларація щодо унікальності текстів роботи
та невикористання матеріалів інших авторів без посилань

Хмелівський Юрій Сергійович

Прізвище, ім'я, по батькові

Інформаційних і прикладних технологій

Факультет

122 «Комп'ютерні науки»

Шифр і назва спеціальності

«Сучасні інформаційні технології та програмування»

Освітня програма

ДЕКЛАРАЦІЯ

Усвідомлюючи свою відповідальність за надання неправдивої інформації, стверджую, що подана кваліфікаційна (бакалаврська) робота на тему: «Розробка мобільного додатку з іграми» є написаною мною особисто.

Одночасно заявляю, що ця робота:

- не передавалась іншим особам і подається до захисту вперше;
- не порушує авторських та суміжних прав, закріплених статтями 21-25 Закону України «Про авторське право та суміжні права»;
- не отримувались іншими особами, а також дані та інформація не отримувались у недозволений спосіб.

Я усвідомлюю, що у разі порушення цього порядку моя кваліфікаційна (бакалаврська) робота буде відхилена без права її захисту, або під час захисту за неї буде поставлена оцінка «незадовільно».

дата

підпис